UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/804,516	03/19/2004	Arun Kwangil Iyengar	YOR920040025US1	7509

7590 12/21/2010

Ryan, Mason & Lewis, LLP
90 Forest Avenue
Locust Valley, NY 11560

EXAMINER	
PHUNG, LUAT	
ART UNIT	PAPER NUMBER
2464	

MAIL DATE	DELIVERY MODE
12/21/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/804,516
Filing Date: March 19, 2004
Appellant(s): IYENGAR ET AL.

David E. Shifren
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 21 October 2010 appealing from the Office action mailed 16 March 2010.

(1) Real Party in Interest

The examiner has no comment on the statement, or lack of statement, identifying by name the real party in interest in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The following is a list of claims that are rejected and pending in the application:

Claims 1-25 are pending.

Claims 1-25 have been rejected in an Office Action mailed 16 March 2010.

Claims 1-25 are presented for appeal.

(4) Status of Amendments After Final

The examiner has no comment on the appellant's statement of the status of amendments after final rejection contained in the brief.

(5) Summary of Claimed Subject Matter

The examiner has no comment on the summary of claimed subject matter contained in the brief.

(6) Grounds of Rejection to be Reviewed on Appeal

The examiner has no comment on the appellant's statement of the grounds of rejection to be reviewed on appeal. Every ground of rejection set forth in the Office action from which the appeal is taken (as modified by any advisory actions) is being

maintained by the examiner except for the grounds of rejection (if any) listed under the subheading "WITHDRAWN REJECTIONS." New grounds of rejection (if any) are provided under the subheading "NEW GROUNDS OF REJECTION."

(7) Claims Appendix

The examiner has no comment on the copy of the appealed claims contained in the Appendix to the appellant's brief.

(8) Evidence Relied Upon

2004/0162901	Mangipudi, et al	8-2004
2005/0198200	Subramanian, et al	9-2005
6,112,221	Bender, et al	8-2000
2003/0120705	Chen, et al	6-2003
6,807,156	Veres, et al	10-2004
6,981,029	Menditto, et al	12-2005
6,772,211	Lu, et al	8-2004

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

3. The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

4. Claims 1, 5-9 and 14-17 are rejected under 35 U.S.C. 103(a) as being unpatentable over Mangipudi, et al (2004/0162901); or over Mangipudi, et al in view of Subramanian, et al (US 2005/0198200).

Regarding claims 1 and 17, Mangipudi discloses a method of processing a request to at least one server, and an article of manufacture for processing a request to at least one server, comprising a computer readable medium containing one or more programs which when executed implement the steps of:

a processor receiving the request **(para. 45; router receiving a client request for web content)**; and

the processor to submit the request to the at least one server **(para. 47; all client requests are routed to the server selected as the most available and/or efficient server within each class according to a selected load balancing algorithm)** based on: (i) a quality-of-service (QoS) class assigned to a client from which the request originated **(Fig. 8E, 9; para. 45, 69, 70; Class of Service (ie class) is implemented as a function of the user; user is authenticated and the respective class is assigned)**; (ii) a response target associated with the QoS class **(para. 21, 25, 26, 54; routing by class to meet user's expectations, SLA metrics such as response times fall within committed levels)**; and (iii) an estimated response time associated with the at least one server. **(para. 55; server attributes such as response times of back-end servers are reported to router as input to policy engine)**

In this embodiment Mangipudi teaches to the processor submit the request to the at least one server (para. 47) as recited above; however Mangipudi does not expressly teach the processor *determining when* to submit the request to the server.

However it is well known in the art that scheduling a task refers to the timing of performing that task. For example, **the sole definition for "schedule" in the Microsoft Computer Dictionary, Fifth Edition, is a verb meaning "To program a computer to perform a specified action at a specified time and date."** Accordingly scheduling submission of the request means to submit the request at a specified time and date, i.e., determining when to submit the request. Furthermore Mangipudi

discloses a well known technique of **scheduling HTTP requests by placing them in queues (para. 9, lines 12-13), and the queues are serviced by the request controller based on configured policy such as length of queues, etc. (para. 11)**, i.e., the request controller 11 determines when to service the requests. Thus it would have been obvious to one of ordinary skill in the art at the time of the invention to implement the request controller and queues as disclosed in Mangipudi's Background of the Invention in combination with scheduling the request to the server based on the criteria cited above in order to efficiently manage processing of the web requests.

In an alternate rejection, Subramanian from an analogous art discloses **a web-service facilitator enabling web-service requests to be load-balanced to servers (para. 9, lines 8-10), and a scheduler configured to enable the web-service request to be scheduled for future execution (para. 9, lines 17-18)**, i.e., determining when to submit the request to the server. Thus it would have been obvious to one of ordinary skill in the art at the time of the invention to implement handling of requests as taught by Subramanian in the system of Mangipudi by scheduling the requests for future execution by the servers. The motivation would be to enable such requests to be managed, monitored, and/or tracked so that progress can be determined across the servers **(para. 8).**

Regarding claim 5, Mangipudi further discloses further comprising the step of assigning the response target to the QoS class **(para. 54)**.

Regarding claim 6, Mangipudi further discloses wherein the step of assigning the response target to the QoS class further comprises the step of assigning a response time target to the QoS class. **(para. 54)**

Regarding claim 7, Mangipudi further discloses wherein the step of assigning the response target to the QoS class further comprises the step of assigning a response percentile target to the QoS class. **(para. 48)**

Regarding claim 8, Mangipudi further discloses further comprising the step of estimating the response time associated with the at least one server based on one or more requests sent to the at least one server within a given time period. **(para. 55, 56)**

Regarding claim 9, Mangipudi further discloses further comprising the step of assigning a target response time to a plurality of QoS classes in which lower quality classes are assigned larger response times than higher quality classes. **(para. 38)**

Regarding claim 14, Mangipudi further discloses an apparatus for processing a request to at least one server, comprising:

a memory; **(para. 39)** and

at least one processor coupled to the memory **(para. 39)** and operative to perform the method of claim 1, and is therefore rejected under the same reason set forth in the rejection of claim 1.

In this embodiment Mangipudi does not expressly teach:

wherein scheduling submission of the request to the at least one server comprises determining when to submit the request to the at least one server.

However it is well known in the art that scheduling a task refers to the timing of performing that task. For example, **the sole definition for "schedule" in the Microsoft Computer Dictionary, Fifth Edition, is a verb meaning "To program a computer to perform a specified action at a specified time and date."** Accordingly scheduling submission of the request means to submit the request at a specified time and date, i.e., determining when to submit the request. Furthermore Mangipudi discloses a well known technique of **scheduling HTTP requests by placing them in queues (para. 9, lines 12-13), and the queues are serviced by the request controller based on configured policy such as length of queues, etc. (para. 11)**, i.e., the request controller 11 determines when to service the requests. Thus it would have been obvious to one of ordinary skill in the art at the time of the invention to implement the request controller and queues as disclosed in Mangipudi's Background of the Invention in combination with scheduling the request to the server based on the criteria cited above in order to efficiently manage processing of the web requests.

In an alternate rejection, Subramanian from an analogous art discloses **a web-service facilitator enabling web-service requests to be load-balanced to servers (para. 9, lines 8-10), and a scheduler configured to enable the web-service request to be scheduled for future execution (para. 9, lines 17-18)**, i.e., determining when to submit the request to the server. Thus it would have been obvious to one of ordinary skill in the art at the time of the invention to implement handling of requests as taught by Subramanian in the system of Mangipudi by scheduling the requests for future execution by the servers. The motivation would be to enable such requests to be

managed, monitored, and/or tracked so that progress can be determined across the servers **(para. 8)**.

Regarding claim 15, Mangipudi further discloses wherein the memory and the at least one processor form a scheduler **(Fig. 3, element 200)** that is external to the at least one server **(Fig. 3, elements 206)**.

Regarding claim 16, Mangipudi further discloses wherein the scheduler is a front-end scheduler and the at least one server is a back-end server **(Fig. 3, elements 200, 206; para. 39)**.

5. Claims 2-4, 18-20 and 25 are rejected under U.S.C. 103(a) as being unpatentable over Mangipudi, et al, in view of Subramanian, et al, and further in view of Bender, et al (US 6,112,221), and further in view of Chen, et al (US 2003/0120705).

Regarding claim 2, the combination of Mangipudi and Subramanian discloses all of the subject matter as disclosed previously in this office action except for the following:

further comprising the step of withholding the request from submission to the at least one server when the request originated from a client assigned to a first QoS class to allow a request that originated from a client assigned to a second QoS class to meet a response target associated therewith.

However scheduling of requests based on priority queues according to response target is well known in the art. For example, Bender from the same or similar fields of endeavor discloses a server which employs a pre-emptive setting not continuously

processing a request, but scheduling them according to an earliest deadline first methodology, by alternately processing the request with the earliest deadline first, followed by that with the next earliest deadline, and so on **(col. 4, lines 52-58; col. 5, lines 27-35).** That is, in Bender, requests are scheduled based on a response target, when it is to be completed. Additionally Chen from analogous art discloses request scheduling using priority queues, in which a request in priority two queue is processed after a request in priority one is. **(fig. 1A, elements 52, 56; para. 19-21)** Thus it would have been obvious to the person of ordinary skill in the art at the time of the invention to combine the request processing method of Mangipudi and Subramanian with the pre-emptive scheduling method based on response target of Bender and the priority queues of Chen by scheduling requests into priority queues according to their target times. The motivation for such a combination would have been to ensure proper order of processing requests.

Regarding claims 3 and 4, the combination of Mangipudi, Subramanian, Bender and Chen discloses substantially all of the subject matter as disclosed previously in this office action. Mangipudi further discloses:

determining a throughput of the at least one server ... to increase throughput of the at least one server **(para. 54, traffic being routed to the servers to maintain consistent response times and service level commitments; para. 57, performance of each server is monitored, specifically real-time status of traffic flow to the web server)**, as recited in claim 3; and

monitoring a throughput of the at least one server … to balance throughput and request response times **(para. 54, traffic being routed to the servers to maintain consistent response times and service level commitments; para. 57, performance of each server is monitored, specifically real-time status of traffic flow to the web server)**, as recited in claim 4.

Mangipudi does not explicitly disclose:

reducing a request withhold rate to increase throughput of the at least one server, as recited in claim 3;

varying a request withhold rate to balance the throughput and request response times, as recited in claim 4.

Bender further discloses a server which employs a pre-emptive setting of scheduling requests according to an earliest deadline first methodology **(col. 4, lines 52-58; col. 5, lines 27-35)**, calculates processing time and dead line for each request **(Fig. 2, element 102)**, and continues adjusting estimated processing time **(Fig. 2, element 112)**. Bender further discloses repeatedly calculating the stretch value in scheduling requests to be processed **(col. 7, lines 24+)**. In Bender, the stretch value determines how long the requests remain in queue and when they are to be processed, thus continually updating the stretch value is analogous to the claimed limitation of varying the request withhold rate. Thus it would have been obvious to the person of ordinary skill in the art at the time of the invention to combine the request processing method of Mangipudi and Subramanian with the pre-emptive scheduling method based on response target of Bender and the priority queues of Chen by scheduling requests

into priority queues according to their target times, monitoring the throughput of the servers, and adjusting the pre-emption rate by continually computing the stretch value so that the request can be processed meeting the response time and service level commitments as required by Mangipudi. The motivation for such a combination would have been to ensure proper order of processing requests.

Regarding claims 18 and 25, Mangipudi discloses assigning at least one client to a quality-of-service (QoS) class from among at least two QoS classes **(para. 70)**, assigning a response target to at least one QoS class **(para. 54)**; and estimating at least one response time of the at least one server based on one or more requests sent to the server within a given time period **(para. 55, 56).** Mangipudi further discloses categorizing higher end requests to a specific cluster of servers assigned more resources guarantees priority is given to this class over other classes **(para. 24),** routing by class **(para. 26),** and server is selected based on load balancing algorithm defined for the cluster or class assigned to the request **(para. 46).** I.e., in Mangipudi, requests are associated with a class of service, based on which processing is performed.

Mangipudi discloses all of the subject matter except:

a processor withholding submission of requests associated with a first one of the at least two QoS classes to allow requests associated with a second one of the at least two QoS classes to meet its response target based on the at least one estimated response time.

However scheduling of requests based on priority queues according to response target is well known in the art. For example, Bender from the same or similar fields of

endeavor discloses a server which employs a pre-emptive setting not continuously processing a request, but scheduling them according to an earliest deadline first methodology, by alternately processing the request with the earliest deadline first, followed by that with the next earliest deadline, and so on **(col. 4, lines 52-58; col. 5, lines 27-35).** That is, in Bender, requests are scheduled based on a response target, when it is to be completed. Additionally Chen from analogous art discloses request scheduling using priority queues, in which a request in priority two queue is processed after a request in priority one is. **(fig. 1A, elements 52, 56; para. 19-21)** Thus it would have been obvious to the person of ordinary skill in the art at the time of the invention to combine the request processing method of Mangipudi with the pre-emptive scheduling method based on response target of Bender and the priority queues of Chen by scheduling requests into priority queues according to their target times. The motivation for such a combination would have been to ensure proper order of processing requests.

Claims 19 and 20 are substantial duplicates of claims 3 and 4, respectively, and are therefore rejected under the same reason set forth in the rejection of claims 3 and 4, respectively.

6. Claim 8 is rejected, in an alternative, under U.S.C. 103(a) as being unpatentable over Mangipudi, et al in view of Subramanian, et al, and further in view of Veres, et al (US 6,807,156).

Regarding claim 8, Mangipudi discloses all of the subject matter as previously recited in this office action. Furthermore, Veres from the same or similar fields of

endeavor discloses further comprising the step of estimating the response time **(col. 13, lines 46-47)** associated with the at least one server or applications based on one or more requests sent to the at least one server or applications within a given time period **(time window of measurement as shown in Fig. 2; col. 13, lines 36-47)**. Thus it would have been obvious to the person of ordinary skill in the art at the time of the invention to combine the request processing method of Mangipudi with the response time estimating method of Veres by periodically sending requests to the applications and servers to estimate the response time. The motivation for such a combination would have been to ensure service level agreement based on response time is met.

7. Claims 10-12 are rejected under U.S.C. 103(a) as being unpatentable over Mangipudi, et al in view of Subramanian, et al, and further in view of Menditto, et al (US 6,981,029).

Regarding claim 10, the combination of Mangipudi and Subramanian discloses all of the subject matter as disclosed previously in this office action except for the following:

determining dispatch times for requests from a difference between at least one predicted response time of the at least one server and the target response time corresponding to the QoS class of the request; and

sending requests to the at least one server based on dispatch times.

However Mangipudi discloses using response time as a metric to select a server to meet SLA commitments **(para. 61, 62).** Menditto from the same or similar fields of

endeavor discloses a content gateway making routing decisions based on the request, selecting a server satisfying the request and depending on various factors such as server load, and forwarding the request to the selected server **(col. 3, lines 11-61)**. Thus it would have been obvious to the person of ordinary skill in the art at the time of the invention to combine the request processing method of Mangipudi and Subramanian with the QoS enforcement approach by Menditto by selecting a server that can timely process the request. The motivation for such a combination would have been to ensure service level agreement based on response time is met.

Regarding claim 11, Mangipudi further discloses:

wherein a plurality of applications are running on the at least one server and requests are routed to applications **(Fig. 3, application servers 216; para. 45)**, further comprising the steps of:

estimating response times of applications based on one or more requests sent to the applications within a time period. **(para. 55, 56)**

The combination of Mangipudi and Subramanian does not explicitly disclose:

sending a request to an application whose estimated response time is not greater than a target response time corresponding to the QoS class of the request.

However Mangipudi discloses using response time as a metric to select a server to meet SLA commitments **(para. 61, 62).** Menditto from the same or similar fields of endeavor discloses selecting an optimal server based on a set of rules, defining as producing the quickest response time to the request. **(col. 6, lines 16-40)** Thus it would have been obvious to the person of ordinary skill in the art at the time of the

invention to combine the request processing method of Mangipudi and Subramanian with the QoS enforcement approach by Menditto by selecting a server that can timely process the request. The motivation for such a combination would have been to ensure service level agreement based on response time is met.

Regarding claim 12, the combination of Mangipudi and Subramanian discloses all of the subject matter as recited above except:

further comprising the step of varying a number of requests sent to applications so that estimated response times of applications are not greater than target response times of QoS classes corresponding to requests sent to the applications.

However Mangipudi discloses sending requests to applications to a server based on response time **(para. 61, 62)** Menditto from the same or similar fields of endeavor discloses a content gateway updating policies regarding processing of requests based on service level agreements **(col. 7, lines 1-52)**. Thus it would have been obvious to the person of ordinary skill in the art at the time of the invention to combine the request processing method of Mangipudi and Subramanian with the QoS enforcement approach by Menditto by selecting a server that can timely process the request. The motivation for such a combination would have been to ensure service level agreement based on response time is met.

8. Claim 13 is rejected under 35 U.S.C. 103(a) as being unpatentable over Mangipudi, et al in view of Subramanian, et al and Menditto, et al, and further in view of Lu, et al (US 6,772,211).

Regarding claim 13, the combination of Mangipudi, Subramanian and Menditto discloses all of the subject matter as disclosed previously in this office action except *wherein the at least one server comprises a plurality of servers and each application runs on a different one of the plurality of servers.*

Lu from the same or similar fields of endeavor discloses methods to switch client packets to one server among a group of servers **(col. 4, lines 50-53)** and applications have their own dedicated servers **(col. 5, lines 24-26)**.

Thus it would have been obvious to the person of ordinary skill in the art at the time of the invention to combine the packet processing method of Mangipudi, Subramanian and Menditto with the servers and applications of Lu by implementing the method and each application on a separate server. The motivation for such a combination would have been to modularize the features for scalability and performance.

9. Claims 21-23 are rejected under U.S.C. 103(a) as being unpatentable over Mangipudi, et al, in view of Subramanian, et al and Bender, et al, Chen, et al and Menditto et al.

Claims 21-23 are substantial duplicates of claims 10-12, respectively, and are therefore rejected under the same reason set forth in the rejection of claims 10-12, respectively.

10. Claim 24 is rejected under U.S.C. 103(a) as being unpatentable over Mangipudi, et al in view of Subramanina, et al, Bender, et al, Chen, et al and Menditto et al, and in further view of Lu, et al.

Claim 24 is a substantial duplicate of claims 13 and is therefore rejected under the same reason set forth in the rejection of claim 13.

(10) Response to Argument

1. Regarding claim 1, appellant argues that Mangipudi discloses in what order the requests will be serviced, but Mangipudi does not disclose when to submit the request. (emphasis original)

Examiner's response:

As a recap of the rejection of claim 1, Mangipudi discloses in the Background of the Invention a well known technique of **scheduling HTTP requests by placing them in queues (para. 9, lines 12-13), requests coming from clients to request controller 10, the requests then being sent to one of the connected web servers 14 (fig. 1, web servers 1, 2, 3), admitted requests are queued into high, medium, and low priority queues, and the queues are serviced by the request controller 10 based on configured policy such as length of queues, etc. (para. 11)**, i.e., the request controller 10 determines when to service the requests that are in the queues, or when the requests are submitted to one of the web servers; high priority requests will clearly be sent to a web server to be processed before lower priority requests, and requests at the front of a queue will be sent to a server before those at the end of the

queue. Furthermore it is well known in the art that scheduling a task, e.g., "scheduling HTTP requests" in Mangipudi, refers to the timing of performing that task. For example, **the sole definition for "schedule" in the Microsoft Computer Dictionary, Fifth Edition, is a verb meaning "To program a computer to perform a specified action at a specified time and date."** (emphasis added) Accordingly scheduling an HTTP request means to submit the request at a specified time and date, i.e., determining when to submit the request. Thus it would have been obvious to one of ordinary skill in the art at the time of the invention to implement the request controller and queues as disclosed in Mangipudi's Background of the Invention in combination with scheduling the request to the server based on the claimed criteria as recited above, in order to efficiently manage processing of the web requests.

Clearly Mangipudi discloses when to submit the request, as claimed.

2. Regarding claim 1, appellant further argues:

Paragraph [0011] of Mangipudi describes an entirely different product from that described in paragraph [0047], and the two products cannot be combined without undue experimentation, nor without proper hindsight.

Examiner's response:

In response to applicant's argument that the combination cannot be made without undue experimentation, the examiner recognizes that the prior art can be modified or combined to reject claims as *prima facie* obvious as long as there is a reasonable expectation of success. *In re Merck & Co.*, Inc., 800 F.2d 1091, 231 USPQ

375 (Fed. Cir. 1986); *Ex parte Blanc*, 13 USPQ2d 1383 (Bd. Pat. App. & Inter. 1989). See MPEP 2143.02 sec. I. Applicant has not presented any evidence showing there was no reasonable expectation of success. See MPEP 2143.02 sec. II. In this case, queuing requests which are then scheduled to be sent to a server, i.e., determining when to send the requests, as disclosed in the Background of Invention of Mangipudi, is a fundamental and well-known feature of data communications system, and as such would have been obvious to one of ordinary skill in the art at the time of the invention to incorporate such a request scheduling feature in with scheduling based on the three claimed criteria in order to efficiently manage processing of the web requests.

In response to applicant's argument that the examiner's conclusion of obviousness is based upon improper hindsight reasoning, it must be recognized that any judgment on obviousness is in a sense necessarily a reconstruction based upon hindsight reasoning. But so long as it takes into account only knowledge which was within the level of ordinary skill at the time the claimed invention was made, and does not include knowledge gleaned only from the applicant's disclosure, such a reconstruction is proper. See *In re McLaughlin*, 443 F.2d 1392, 170 USPQ 209 (CCPA 1971).

3. Regarding claim 1, appellant further argues that Subramanian fails to remedy Mangipudi so as to teach or suggest determining when to submit the request to the at least one server based on: (i) a quality-of-service (QoS) class assigned to a client from

which the request originated; (ii) a response target associated with the QoS class; and (iii) an estimated response time associated with the at least one server.

Examiner's response:

Applicant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references.

It is noted that Subramanian is not relied on to teach the three criteria for scheduling requests, which are taught by Mangipudi as recited in the office action. As a recap of the rejection of claim 1, Subramanian from an analogous art discloses **a web-service facilitator enabling web-service requests to be load-balanced to servers (para. 9, lines 8-10), and a scheduler configured to enable the web-service request to be scheduled for future execution (para. 9, lines 17-18)**, i.e., determining when to submit the request to the server. Thus it would have been obvious to one of ordinary skill in the art at the time of the invention to implement handling of requests as taught by Subramanian in the system of Mangipudi, specifically scheduling the requests for future execution by the servers, as suggested by Subramanian, using the criteria taught by Mangipudi. The motivation would be to enable such requests to be managed, monitored, and/or tracked so that progress can be determined across the servers **(Subramanian, para. 8).**

4. Regarding claims 5 and 6, appellant argues that there is no teaching or suggestion directed to assigning a response target to any particular QoS class, and specifically, assigning a response time target to the QoS class.

Examiner's response:

As shown in the office action, Mangipudi further discloses traffic being routed to the servers, as a function of class, to maintain consistent response times and service level commitments (para. 54). Thus it would have been obvious to one of ordinary skill in the art that routing traffic as a class to maintain committed service level and response times is performed by assigning a response time target to a class during processing of requests. Mangipudi thus teaches the claimed limitation.

5. Regarding claim 7, appellant argues that there is no teaching or suggestion directed to assigning a response percentile target to any particular QoS class.

Examiner's response:

As shown in the office action, Mangipudi further discloses traffic being routed to the servers, as a function of class, to maintain consistent response times and service level commitments (para. 54), and weighted percentage load balancing option is used, e.g., certain servers will receive higher percentage of connections at any one time. In Mangipudi, requests are routed as a function of class based on percentage of connections. Mangipudi thus implicitly discloses assigning percentile targets to achieve the committed service level. Thus it would have been obvious to one of ordinary skill in the art that routing traffic as a class to maintain committed service level is performed by

assigning a percentile target to a class during processing of requests. Mangipudi thus teaches the claimed limitation.

6. Regarding claim 9, appellant argues that there is no teaching or suggestion directed to assigning a target response time to a plurality of QoS class in which lower quality classes are assigned larger response time than higher quality classes.

Examiner's response:

As shown in the office action, Mangipudi further discloses traffic being routed to the servers, as a function of class, to maintain consistent response times and service level commitments (para. 54), and assigning more resources to a cluster to support higher class requests guarantees that more resources are available to this class (para. 38). In Mangipudi, more resources are available to a class and requests are processed to meet committed response times. Mangipudi thus implicitly discloses assigning requests corresponding to a higher class needing a quicker response times, or shorter response times, and vice versa. Thus it would have been obvious to one of ordinary skill in the art that routing traffic as a class to maintain committed service level is performed by assigning a larger response times to a lower quality class during processing of requests. Mangipudi thus teaches the claimed limitation.

7. Regarding claim 2, appellant argues that Bender does not teach or suggest any technique which involves withholding submission of requests to the server and that

Bender does not schedule jobs based on a "response target" associated with a particular QoS class.

Examiner's response:

As shown in the rejection of claim 2, Mangipudi further discloses categorizing higher end requests to a specific cluster of servers assigned more resources guarantees priority is given to this class over other classes **(para. 24),** routing by class **(para. 26),** and server is selected based on load balancing algorithm defined for the cluster or class assigned to the request **(para. 46).** I.e., in Mangipudi, requests are associated with a class of service, based on which processing is performed.

I.e., Mangipudi schedules jobs based on a "response target" associated with a particular QoS class and withholding submission of requests to the server.

Bender from the same or similar fields of endeavor discloses a server which employs a pre-emptive setting not continuously processing a request, but scheduling them according to an earliest deadline first methodology, by alternately processing the request with the earliest deadline first, followed by that with the next earliest deadline, and so on **(col. 4, lines 52-58; col. 5, lines 27-35).** That is, in Bender, a request is scheduled based on a response target, when it is to be completed. Additionally Chen from an analogous art discloses request scheduling using priority queues, in which a request in priority two queue is processed after a request in priority one is. **(fig. 1A, elements 52, 56; para. 19-21)** Thus it would have been obvious to the person of ordinary skill in the art at the time of the invention to combine the request processing method based on class of service of Mangipudi with the pre-emptive scheduling method

based on response target of Bender and the priority queues of Chen by determining when to submit requests using priority queues and according to their target times. The motivation for such a combination would have been to ensure proper order of processing requests.

Clearly the combination of Mangipudi, Bender and Chen discloses the claimed limitations of withholding submission of requests to the server and determining when to submit jobs based on a "response target" associated with a particular QoS class.

8. Regarding claims 3 and 4, appellant argues that Bender is silent as to reducing a request withhold rate, much less doing so to increase throughput (claim 3), and that Bender is silent as to varying a request withhold rate, much less doing so to balance throughput and response times (claim 4).

Examiner's response:

As recited in the office action, Mangipudi discloses traffic being routed to the servers to maintain consistent response times and service level commitments (para. 54), and performance of each server is monitored, specifically real-time status of traffic flow to the web server (para. 57). I.e., performance, or throughput, of the servers, along with response times, are commitments to be monitored and achieved, as claimed. Furthermore as shown in the rejection of claims 3 and 4, Bender discloses a server which employs a pre-emptive setting of scheduling requests according to an earliest deadline first methodology **(col. 4, lines 52-58; col. 5, lines 27-35)**, calculates processing time and dead line for each request **(Fig. 2, element 102)**, and continues

adjusting estimated processing time **(Fig. 2, element 112)**. Bender further discloses repeatedly calculating the stretch value in scheduling requests to be processed **(col. 7, lines 24+)**. In Bender, the stretch value determines how long the requests remain in queue and when they are to be processed, thus continually updating the stretch value is analogous to the claimed limitation of varying the request withhold rate. Thus it would have been obvious to the person of ordinary skill in the art at the time of the invention to combine the request processing method of Mangipudi and Subramanian with the pre-emptive scheduling method based on response target of Bender and the priority queues of Chen by scheduling requests into priority queues according to their target times, monitoring the throughput of the servers, and adjusting the pre-emption rate by continually computing the stretch value so that the request can be processed meeting the response time and service level commitments as required by Mangipudi. The motivation for such a combination would have been to ensure proper order of processing requests.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/Luat Phung/

Examiner, Art Unit 2464

Conferees:

/Ricky Ngo/

Supervisory Patent Examiner, Art Unit 2464

/Huy D Vu/
Supervisory Patent Examiner, Art Unit 2461